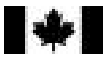
Review of Network-Based Approaches for Privacy

Song, Ronggong; Korba, Larry



National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC-CNRC

*Review of Network-based Approaches for Privacy. **

Song, R. and Korba, L.

May 2002

* published in: Proceedings of the 14th Annual Canadian Information Technology Security Symposium, Ottawa, ON. May 13-17, 2002. NRC 44905.

Canada

Review of Network-Based Approaches for Privacy

Ronggong Song Larry Korba
Institute for Information Technology,
National Research Council of Canada
E-mail: {Ronggong.Song, Larry.Korba}@nrc.ca

Abstract

We first provide an overview of the better-known network approaches for assuring anonymity and privacy over networks. These approaches include anonymous communication networks such as MIX network, Onion Routing network, Crows system and Freedom network. We also examine peer-to-peer networks such as OpenPrivacy and JXTA. We then analyze their effectiveness and the possible attacks to these networks based on traffic analysis techniques. We also discuss their implementation and design issues for network privacy, and some problems engendered by firewalls, dynamic IP, NAT and VPN. In the conclusions we summarize the results and describe the research opportunities in this domain.

Keywords: Privacy, anonymity, anonymous communication networks, peer-to-peer network systems.

1. Introduction

One of the prime impediments to user participation to e-business activities is the fear of violation of individual privacy. Governments around the world have introduced or are building legislation that places requirements upon the way in which personal information is handled. Meeting the privacy challenges of the privacy principles behind this legislation is difficult. The Data Protection principle requires that personal data must be protected so that it cannot be used by the unauthorized. Data must be protected when stored and when transmitted over networks.

In attempt to provide some technical solutions to fill the apparent privacy void for computer network information exchange, several network-based privacy-enhancing technologies have been developed in recent years. In this document we provide an overview of the better-known network approaches for assuring anonymity and privacy over networks. The key approaches we discuss include: MIX network [2, 10], Onion Routing network [3, 4, 8], Crowds system [9] and Freedom network [11]. We also briefly examine some peer-to-peer networks such as OpenPrivacy [5] and JXTA [1, 7].

The goal of these approaches is to protect users against traffic analysis since an adversary can monitor and compromise certain parts of the systems to be able to match a message sender with the receiver. This is an important property for protecting the identity of communication partners in an e-business context for confidentiality purposes. It is also a property desired by Internet users who want to keep their Internet lives and relations private. While these approaches offer some possibilities for providing anonymity and privacy, most of them are vulnerable to traffic analysis attacks. We introduce the general methods of traffic analysis attacks, and discuss the techniques used to protect against these attacks, and then give a comparison of the network-based approaches against these attacks.

We discuss the implementation issues that prevent the ubiquitous deployment of these techniques. As well, we based on these findings, we outline directions for further research. The purpose of this paper is not to present new results, but rather to encourage further research in the area of Internet privacy protection and give an overview of network-based technologies for privacy.

The rest of the paper is organized as follows. Some network-based approaches are briefly reviewed in the next section. In Section 3, the possible threats of these networks are analyzed and compared. In Section 4, the implementation and

design issues of these networks are concisely described. In Section 5, some concluding remarks and directions proposing important directions for further research.

2. Review of Network-based Approaches for Privacy

Privacy requirements have been recognized for many years. TCP over IP version 4 is designed to allow computers to easily interconnect and to assure that network connections will be maintained even when various links may be damaged. This same versatility makes it rather easy to compromise data privacy in networked applications. For instance, networks may be sniffed for unencrypted packets, threatening the confidentiality of data. Research and development however, have led to techniques that provide varying levels of private communication between parties. In this section we concisely describe some of the more commonly known network privacy technologies.

2.1 MIX Network

In order to enable unobservable communication between users of the Internet, David Chaum [2] introduced MIX-networks in 1981. A MIX network takes a list of values as input, and outputs a permuted list of function evaluations of the input items, without revealing the relationship between input and output elements.

A MIX-network is composed of MIX nodes. A MIX node is a processor that receives a certain number of messages, modifies them using some cryptographic transformation and outputs them in a random order in such a way that one cannot correlate messages that "come in" with messages that "go out". MIX nodes can be used to prevent traffic analysis in roughly the following manner.

(1) The message will be sent through a series of MIX nodes, say $i_1, i_2, \ldots, i_d$. The user encrypts the message with node i_d's key, encrypts the result with node i_{d-1}'s key and so on with the remaining keys.
(2) The MIX nodes receive a certain number of these messages, which they decrypt, randomly reorder and send to the next nodes in the routes.

Each MIX node in the network knows only the previous and next node in a received message's route. Hence, unless the route only goes through a single node, compromising a MIX node doesn't trivially enable an attacker to violate sender-recipient privacy. When using only one MIX, one must rely upon security of that node completely. Usually several MIXes are used in a chain. In this manner, any single MIX does not have enough information needed to reveal communication relations. At worst, a MIX may only know either sender or receiver.

2.2 Onion Routing

The primary goal of Onion Routing [3, 4, 8] is to provide strongly private communications in real time over a public network with reasonable cost and efficiency. A secondary goal is to provide anonymity to the sender and receiver, so that the responder may receive messages but be unable to identify the sender, even though the responder may be able to reply to those messages.

In onion routing, initiating applications make connections through a sequence of onion routers instead of making socket connections directly to responding machine. Onion routers are computer programs that perform application-layer routing for the network. The onion routing network allows the connection between the initiator and responder to remain anonymous, and is thus called an anonymous connection. Onion Routing builds anonymous connections within a network of onion routers, which are, roughly, real-time Chaum MIXes. While Chaum's MIXes could store messages for an indefinite amount of time waiting to receive an adequate number of messages to mix together, a Core Onion Router is designed to pass information in real time, which limits mixing and potentially weakens the protection. Just as large volumes of traffic improve the protection of real time MIXes, the large traffic is vital to strengthen Onion Router networks.

Onion routers in the network are connected by longstanding socket connections. Anonymous connections through the application layer onion routing network are multiplexed over these longstanding connections. For any anonymous connection, the sequence of onion routers in a route is strictly defined at connection setup. However, each onion router can only identify the previous and next hops along a route. Data passed along the anonymous connection appears differently at each onion router, so data cannot be tracked en route.

With Onion Routing, an initiating application makes a socket connection to an application specific proxy on some onion router. That proxy defines a route through the onion routing network by constructing a layered data structure called an onion. The onion is sent through the network to set up the route. Each layer of the onion defines the next hop in a route. An onion router that receives an onion peels off its layer, identifies the next hop, and sends the embedded onion to that onion router. Once the anonymous connection is established, data can be sent in both directions. The initiator's onion proxy receives data from an application, breaks it into fixed sized cells, and encrypts each cell with the key shared with the last node, and the result is then encrypted with the key shared with the penultimate node, and so on. As a cell of data moves through the anonymous connection, each onion router removes one layer of encryption, so that the data emerges as plaintext from the final onion router in the path. The responder proxy regroups the plaintext cells into the data stream originally submitted by the application and forwards it to the destination. For data moving backward, from the recipient to the initiator, this process occurs in the reverse order, with the responder proxy breaking the traffic into cells, and successive onion routers encrypting it using different algorithms and keys than the forward path. In this case the initiator's proxy decrypts the data multiple times, regroups the plaintext cells, and forwards them to the application.

2.3 Freedom Network

The Freedom Network [11] is composed of a set of nodes called Anonymous Internet Proxies, which run on top of the existing Internet infrastructure. It uses layers of encryption to allow a Freedom user to engage in a wide variety of pseudonymous activities, hiding the user's real IP address, email address, and other identifying information from eavesdroppers and active attempts to violate the user's privacy. Users are encouraged to create pseudonyms for each area of activity in which they want to preserve their privacy.

The main components of the Freedom Network are Freedom Clients and Freedom Servers. The client uses a route creation protocol to set up a communication channel through the Freedom Network. This protocol enables the client to share two secret keys for bi-direction with each Freedom Server node, as well as to tell each node what the previous and next nodes are in the route. Each node then sets a pair of Anonymous Connection Identifiers, which associate next and previous nodes with the route, and ends up knowing only what the next and previous nodes are on the certain route.

There are two different operations with Onion Routing and MIX-networks in Freedom Network. One is that although it also uses the nested encryption channel, its encrypted object is the whole IP packet. Another is that when the last node receives a packet, it replaces the missing IP source address with a special IP address called the wormhole IP address.

2.4 Crowds System

Reiter and Rubin [9] propose a lighter weight alternative to MIXes. Their system is called Crowds System that is based on a very different principle, and can be seen as a P2P (peer-to-peer) relaying network in which all participants forward messages. The goal of Crowds is to make browsing anonymous, so that information about either the user or what information he or she retrieves is hidden from Web servers and other parties. Crowds prevents a Web server from learning any potentially identifying information about the user, including the user's IP address or domain name. Crowds also prevents Web servers from learning a variety of other information, such as the page that referred the user to its site or the user's computing platform.

Crowds consists of a number of network nodes that are run by the users of the system. The approach is based on the idea of "blending into a crowd", i.e., hiding one's actions with the actions of many others. To execute web transactions in this model, a user first joins a **crowd** of other users. The user's initial request to a web server is first passed to a random member of the crowd. That member can either submit the request directly to the end server or forward it to another randomly chosen member. In the latter case the next member independently chooses to forward or submit the request. Finally, the request is submitted by a random member, thus preventing the end server from identifying its true initiator. Even crowd members cannot identify the initiator of the request, since the initiator is indistinguishable from a member that simply passed on a request from another. Unlike the above anonymous networks, Crowds doesn't use a nested encryption channel. This leads to more efficient performance.

2.5 OpenPrivacy

OpenPrivacy [5] is a distributed peer-to-peer network. It provides a framework for building intercommunicating systems that supports the concept of reputation through opinion accumulation. Opinion, which can be attached to any object such as pseudonyms, purchase histories, physical objects and reputation servers, are pervasive and directly affect every aspect of OpenPrivacy-enabled systems.

OpenPrivacy creates a networked peer-to-peer platform enabling Open Privacy Providers to provide people with complete control over and protection of their personal information. The platform provides (1) storage, unique naming, indexing and retrieval mechanisms for profiles, (2) privacy – a user's identity cannot be determined from their profile information, and (3) security – a user can determine how their profile is to be used, explicitly, permitting some uses and denying others. The platform also enables applications that will afford the user many direct benefits without the loss of privacy or fear that their data is being used inappropriately.

A set of reputation services such as pseudonym (nym), bias and reputation calculation engine, form the cornerstone of the OpenPrivacy framework. These services provide a standard opinion and reputation framework that can be used by any community, supporting a large variety of mechanisms to create, use and calculate results from accumulated opinions, bias and reputations.

The reference applications for OpenPrivacy include: Sierra – a reference implementation for the components of an OpenPrivacy reputation management framework resulting in a compete reputation management system, Talon – a flexible component system designed to incorporate Sierra as part of its component factory mechanism, and Reptile – a decentralized peer-to-peer application that has a flexible network plug-in infrastructure. Reptile will operation on multiple P2P networks such as JXTA, Freenet, etc. It also has a privacy and reputation-enhanced Internet portal to keep a user's profile anonymous and integrate the Sierra Reputation Framework.

2.6 JXTA

JXTA [1, 7] is a network programming and computing platform designed to enable a wide range of distributed computing applications. Sun Microsystem, the developer of JXTA, states that it overcomes the limitations found in many of today's P2P applications, and offers a set of simple, small and flexible mechanisms that is purported to support P2P computing on any platform, anywhere and at any time.

JXTA breaks down a typical P2P software stack into three layers: core layer, service layer and application layer. The core layer at the bottom provides core support such as peer establishment and communication management for peer-to-peer services and applications. The service layer in the middle deals with higher-level concepts such as indexing, searching and file sharing. At the top is the application layer, such as emailing, auctioning, and storage systems.

JXTA does not yet directly deal with anonymity issues. JXTA assumes the user's anonymity should be ensured through external applications such as naming services and pseudonym services. JXTA platform is independent of the anonymous solution chosen by the particular applications.

3. Analysis of Anonymous Communication Networks

We first describe general traffic analysis attacks on anonymous communication networks, and particularly focus upon the attacks that may reduce the privacy of the individuals. These attacks include message coding attack, timing attack, message volume attack, flooding attack, intersection attack, communication pattern attack, collusion attack and denial of service attack, etc. We then give a comparison of the above networks against these attacks.

3.1 General Traffic Analysis Attacks

- ***Message Coding Attack:*** An attacker can easily link and trace some messages if the messages do not change their coding during transmission. Most anonymous systems, for example, MIX-network, Onion Routing and Freedom network, use the nested layers of encryption technique against this attack. Crowds System provides protection against this attack with a certain probability from insiders and in any case from outsiders. Note that the link-to-link encryption between nodes is not sufficient in order to prevent insider attacks.

- ***Timing Attack:*** An attacker can observe the set of messages coming into the network and the set of message going out of it, to obtain some useful route timing information by correlating the messages in the two sets. If the different routes that can be taken require different amounts of time, the system could be vulnerable to timing attacks. The attacker having access to one of the communicating parties might be able to infer which route is taken by simply computing the round trip time.

  The earlier MIX-network provides protection against this attack by using a high delay between each node. Recently, it has been improved by means of dummy traffic and a chop-and-slice algorithm [10]. But a sophisticated attacker may be able to detect timing coincidence such as the near simultaneous opening of connections. Timing coincidences are very difficult to overcome without wasting network capacity, especially when real-time communication is important.

- ***Communication Pattern Attack:*** An attacker may find out a lot of useful information simply by looking at the communication patterns when users send and receive messages. For example, when one of the communicating participants sends the message, the other is usually silent. The longer the attacker can observe this type of communication synchronization, the less likely it's just an unrelated random pattern.

  A passive adversary can mount this attack by monitoring the entry and exit nodes. Law enforcement officials might be quite successful mounting this kind of attack as they often have a-priori information. They usually have a hunch that two parties are communicating and just want to confirm their suspicion.

  The communication pattern attack is one of most dangerous attacks and is very difficult to model in a rigorous manner. The problem is that real-world users don't behave like those in the idealized model. This attack is particularly effective for real-time, interactive communication.

- ***Packet Volume and Counting Attack:*** An attacker can observe the amount of transmitted data (e.g. the message length, number of messages). This can be accomplished by sniffing packets on any router in the communication path between the sender and the first node. Thus, a global observer is able to associate a communication relation to a certain client and server. The attacker could distinguish the messages sent by the node, if they ere of different size. Because the size of the sent message is the same as the size of the received message, the attacker could correlate them.

  One way of defending against such an attack is to use constant link padding. In this case, the traffic between any two nodes consists of the same-sized packets per time unit. Recently, some networks have been improved against this attack by means of dummy traffic, chop-and-slice algorithm, and traffic shaping [13], etc.

The packet counting and communication pattern attacks can be combined to get a message frequency attack. They are sometimes referred to as traffic shaping attacks and are usually dealt with by imposing rigid structure on user communications.

- ***Message Delaying Attack:*** An attacker can withhold messages until he can obtain enough resources or until the network becomes easier to monitor or to see if the possible recipient receives other messages, etc. In view of this attack, it makes sense to have the nodes verify authenticated timing information.

- ***Message Tagging Attack:*** An active internal attacker, who has control of the first and last node in a message route, can tag messages at the first node in such a way that the exit node can spot them. Since the entry node knows the sender and the exit node the recipient, the system is broken.

  An active external attacker can mount a slight variant of this attack if the messages don't have a rigid structure. This attack has many similarities with subliminal channels [12]. This observation forms the basis of some of the following variations:

  Shadow Messages: If the attacker sends messages that follow the same path as the message being followed, it can easily transmit some information to the output. For example, the attacker can just replay the message in such a way so as to spot it leaving the anonymous network.

  Message Delaying: The attacker can delay messages to obtain some information. These delays can presumably be detected.

  Broadcast: The attacker can broadcast messages notifying his accomplices that a particular message has entered the network. This isn't a particularly powerful attack but it could be virtually impossible to detect.

  A solution to this problem is to make it difficult to tag messages. The techniques that can be used to do this depend on the implementation.

- ***Flooding Attack:*** An attacker may flood a system in order to separate a certain message. For example, if the nodes wait till they have *n* messages before flushing, the attacker can send *n-1* messages and easily associate messages leaving the node with those having entered.

  Dummy traffic can make things a bit more difficult for the attacker since he can't distinguish them from legitimate messages. Unfortunately, if dummy traffic is implemented such that it is only used in specific instances, an attacker then has the opportunity to choose his messages so that dummy traffic will not be used.

  Another potential solution is to check the identity of *n* users, ensuring that a single user is not able to flood a node with *n-1* messages in order to trace the remaining single message. However, in the existing Internet, secure identity management is not available. Thus, the attacker can fake different identities in order to simulate different users. In a practical system it has to be ensured that a message is authenticated. In order to remain anonymous, some blind signature or pseudonym credential techniques can be used for this. Unfortunately, this entails authenticating each message and detecting flooding attempts, which could be computationally infeasible.

- ***Intersection Attack:*** An attacker may trace some users by observation over a long period because of the on-line/off-line periods of the users or a special distinguishable behavior. For example, the typical user usually queries the same web sites in different sessions. By performing an operation similar to an intersection on the sets of active users at different times it is probable that the attacker can gain interesting information. The intersection attack is a well-known open problem and seems extremely difficult to solve in an efficient manner. Dummy traffic may make this attack somewhat harder but does not prevent it.

- ***Collusion Attack:*** A corrupt coalition of users or parts of the system may be able to trace some users. Currently, most anonymous networks can provide protection against *k-1* collusion of *k* nodes. These networks all are distributed systems since no central system can easily provide protection against a corrupt insider.

- ***Denial of Service Attack:*** An attacker may obtain some information about the routes used by certain users by rendering some nodes inoperative.

  In general most anonymous networks consisting of many nodes may be sufficiently robust so that compromising one or two nodes may not provide any information about the data transferred or the parties involved. However, a single compromised routing node can destroy connections or stop forwarding messages, resulting in a denial of service attack. As a result, attacks on routing nodes are detectable. More difficult to detect are active attacks wherein the attackers substitute wrong messages in a system that uses intermediaries to forward unprotected information.

- ***Replay Attack:*** An attacker, who observes the incoming and outgoing messages, would send the same message to the node again, and would determine which message is sent twice from the node.

  Usually, this is achieved by different techniques in different systems. One technique is to use a serial number and a database in which a node stores the serial number of every processed message. If a new message arrives, the node first checks if its serial number isn't already stored in this database. Another is to use expiration times. Due to poorly synchronized clocks, the vulnerability in this system is a denial of service attack instead of a replay attack.

In addition, there also exist some special attacks [6] for each anonymous communication network. We do not describe them here.

3.2 Comparison of Anonymous Communication Networks

Based on the above attacks, a comparison of the network-based approaches in Section 2 for privacy protection functions is described in Table 1. Since the privacy protection techniques are improved from time to time, the following comparison is based on the current techniques implemented or proposed in these approaches.

Table 1: Comparison of the network-based approaches against traffic analysis attacks.

	Message coding attack	Timing attack	Comm. pattern attack	Packet volume and accounting attack	Message delaying attack	Message tagging attack	Flooding attack	Intersection attack	Collusion attack	Denial of service attack	Replay attack
MIX-network	✓	✓	P	✓	✓	×	P	×	✓	P	✓
Onion Routing	✓	✓	P	✓	✓	×	×	×	✓	×	✓
Freedom	✓	✓	P	✓	✓	×	×	×	✓	✓	✓
Crowds	P	P	×	×	×	×	×	×	P	×	×
OpenPrivacy	×	×	×	×	×	×	P	×	P	P	P
JXTA	×	×	×	×	×	×	×	×	×	×	×

✓ – The system provides protection against this attack.
P – The system provides partial protection against this attack.
× – The system does not provide protection against this attack, or does not consider this protection.

4 Implementation and Design Issues

We first describe general implementation and design issues including some problems engendered by firewalls, NAT, and secrity protocols such as SSL and IPSec, and then give a comparison of these approaches based on their implementation functions.

4.1 General Implementation Issues

- ***Anonymity and Pseudonymity:*** Probably the most important design issue is that of anonymity versus pseudonymity. The key advantages of both anonymity and pseudonymity are as follows.

  Anonymity: Provides better security than pseudonymity since it prescribes transmission paths that are random and never used twice.

  Pseudonymity: Provides a best solution for privacy protection and accountability. Since pseudonyms have a persistent nature, long-term relationship, reputation and trust can be cultivated. Thus, pseudonym-based business models also are more attractive than anonymity based ones. In addition, abuse control is easier to deal with when pseudonyms are used, and authentication is easier.

- ***Routing:*** Route finding is another important issue for the network-based approaches for privacy protection. Currently, most anonymous networks choose the routing nodes randomly. For large Internet-based systems especially, having the user choose the nodes randomly doesn't appear to be a viable option. Thus, creating good network topologies and route finding algorithms with respect to security and efficiency is not trivial.

- ***NAT and Firewall:*** The wide spread use of NAT and firewalls presents severe problems for most network-based approaches for privacy protection. This is because a node outside a firewall or a NAT cannot discover nodes inside the firewall or the NAT gateway. This situation may be circumvented if the system administrator gives a special set up on the firewall or the NAT gateway. This is far from an ideal solution, leaving this to be an active research area.

- ***SSL and IPSec:*** SSL and IPSec protocols have become the de facto standard approaches for secure communication over the Internet. But some above network-based approaches do not support them. This would severely affect the use of the systems.

4.2 Comparison of Implementation Issues

Based on the above implementation issues, a comparison of the network-based approaches is described in Table 2. The following comparison also is based on the current techniques implemented or proposed in these approaches.

Table 2: Comparison of the network-based approaches for their implementation issues.

	Anonym and pseudonym	Routing	NAT	Firewall	SSL	IPSec
MIX-network	P	P	×	P	✓	×
Onion Routing	×	×	×	P	×	×
Freedom	Pseudonym	✓	×	P	✓	×
Crowds	×	✓	×	×	×	×
OpenPrivacy	Pseudonym	✓	P	×	×	×
JXTA	×	✓	P	P	✓	×

✓ – The system supports this function.
P – The system partial supports this function.
× – The system does not support this function, or does not consider this function.

5 Further Research

The most important direction for further research in this field should be in the areas of: prevention of attacks implementation issues, and private community building techniques. The general approach when developing any network security protocol is to find and rigorously analyze as many attacks as possible in an attempt to immunize our protocols against these attacks, or detect when the attack is mountable and take the appropriate measures. Unfortunately this may be very difficult or impossible since it is not at all clear what a real-world adversary would do. One need only consider the lack of preparedness for the number of distributed denial of service attacks on web services that occurred within the last few years to realize the challenge of preparing for every potential eventuality. Another example is the e-mail attachment security issue that involved "hidden" vulnerabilities "or features" in email clients that were exploited to produce e-mail havoc.

When considering implementation issues, another important limitation is that the current developed network privacy solutions are not at the IP layer. This means that some important security protocols such as SSL and IPSec are not available in these networks. Additionally, packet loss in some networks such as MIX-network and Onion Routing, cause a problem of network backlog and cascading retransmits, since the nodes talk to each other via TCP. The end-to-end connection assurance under TCP is not available and must be built into the network privacy approaches in higher layers of the IP stack. Thus, an anonymous Internet at the IP layer becomes a desideratum to enable broad-based support of any TCP/IP applications.

Privacy protection and accountability must be considered in the development of new e-business applications. In particular, private credentials and trust mechanisms will be important developments. These are currently in rudimentary form, requiring further development for widespread deployment to meet the emerging needs in e-business. This is especially cases where intelligent agent solutions are being developed. Trust mechanisms will be an important enabling technology facilitating the negotiation of business, security and privacy policies, potentially increasing the speed and security with which e-business may be performed.

6 Conclusion

This document reviews the various solutions that have been proposed and implemented for network privacy. We have also analyzed these approaches based upon the threats from different types of attacks.

While exposures exist in these approaches, it is clear that:

(1) Some of the attacks require significant resources to launch and maintain. This means that for the most part, these approaches are reasonably secure. As such one or several of the previous developments in this area may be used to provide network-level anonymity.
(2) There are several research opportunities in the development of improved privacy networks.

In addition, with the development application of e-business approaches, it is clear that private credential and reputation management approaches will require research and development in order to promote wide-spread acceptance.

Acknowledgments

We would like to thank all members of IIT at the NRC of Canada for their support towards our R&D projects in Privacy Protection. We also thank our European Community partners in the Privacy Incorporated Software Agent (PISA), an IST-EU Fifth Framework Project.

References

[1] B.Traversat and M.Abdelaziz. Project JXTA Virtual Network. Sun Microsystems, Inc., Feb. 2002. Available at http://www.jxta.org/project/www/docs/JXTAprotocols.pdf.

[2] Chaum. Untraceable Electronic Mail, Return Address, and Digital Pseudonyms. Communications of the ACM, vol.24 no.2, pages 84-88, 1981.

[3] D.Goldschlag, M.Reed and P.Syverson. Onion Routing for Anonymous and Private Internet Connections. Communication of the ACM, vol.42, no.2, pages 39-41, 1999.

[4] D.Goldschlag, M.Reed and P.Syverson. Hiding Routing Information. In R.Anderson, editor, Information Hiding: First International Workshop, Volume 1174 of Lecture Notes in Computer Science, pages 137-150, Springer-Verlag, 1996.

[5] F.Labalme and K.Burton. Enhancing the Internet with Reputations. An OpenPrivacy White Paper, Mar. 2001. Available at http://www.openprivacy.org/papers/200103-white.html.

[6] J.Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H.Federrath, editor, Anonymity 2000, Volume 2009 of Lecture Notes in Computer Science, pages 10-29, Springer-Verlag, 2000.

[7] L.Gong. Project JXTA: A Technology Overview. Sun Microsystems, Inc., Apr. 2001. Available at http://www.jxta.org/project/www/docs/TechOverview.pdf.

[8] M.Reed, P.Syverson and D.Goldschlag. Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communications, vol.16, no.4, pages 482-494, May 1998.

[9] M.Reiter and A.Rubin. Anonymous Web Transactions with Crowds. Communications of the ACM, vol.42, no.2, pages 32-48, 1999.

[10] O.Berthold, H.Federrath and S.Kopsell. Web MIXes: A System for Anonymous and Unobservable Internet Access. In H.Federrath, editor, Anonymity 2000, Volume 2009 of Lecture Notes in Computer Science, pages 115-129, Springer-Verlag, 2000.

[11] P.Boucher, A.Shostack and I.Goldberg. Freedom Systems 2.0 Archtecture. December 2000. Available at http://www.freedom.net/info/whitepapers/Freedom_System_2_Architecture.pdf.

[12] G.J.Simmons. The history of subliminal channels. IEEE Journal on Selected Area in Communications, vol. 16, no.4, pages 452-462, May 1998.

[13] A.Back, U.Moller and A.Stiglic. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems. In I.S.Moskowitz, editor, IH 2001, Volume 2137 of Lecture Notes in Computer Science, pages 245-257, Springer-Verlag, 2001.

Biography

Ronggong Song received his B.Sc degree in mathematics in 1992, M.Eng degree in computer science in 1996, Ph.D. in network security from Beijing University of Posts and Telecommunications in 1999. He had employed as Network Planning Engineer at Telecommunication Planning Research Institute of MII, P.R.China, and Postdoctoral Fellow at University of Ottawa, Canada. Now, he is working at NRC of Canada. His research interests are privacy protection, network security, e-commerce, IP mobility and QoS.

Larry Korba is the group leader of the Network Computing Group of the National Research Council of Canada in the Institute for Information Technology. He is currently involved in several projects related to security and privacy.. His research interests include privacy protection, network security, and computer supported collaborative work.